Online Edition
ISSN 2066-8562

Annals of the Academy of Romanian Scientists
Series on Science and Technology of Information
Volume **6**, Number **2**/**2013**

A SYSTEM FOR SONIFICATION OF CHAT CONVERSATIONS

Alexandru CĂLINESCU[1], Ştefan TRAUŞAN-MATU[2]

Abstract. *This paper presents the MusicXML Creator software system that generates an audible representation (a 'sonification') of a chat conversation starting from the polyphonic model introduced by the second author. The obtained musical composition highlights how participants interact and how discussion topics are alternated. The main purpose of the paper is to present how the implemented software system materializes the polyphonic model and analysis method of Computer-Supported Collaborative Learning chat conversations.*

Keywords: Sonification, Computer-Supported Collaborative Learning, polyphony, natural language processing, music composition, polyphonic model, discourse analysis

1. Introduction

This paper presents the development of a software system that generates an audible representation (a 'sonification') of a chat conversation starting from the polyphonic model. The musical composition obtained highlights how participants interact and how discussion topics are alternated.

The main purpose of the paper is to present how the implemented software system materializes the polyphonic model and analysis method of Computer-Supported Collaborative Learning (CSCL) instant messenger (chat) conversations [1, 2].

The polyphonic model considers that the analysis of the degree of contribution and collaboration in CSCL chats can be done starting from an analogy with polyphonic music, in which several threads (voices) enter in inter-animation processes along both the longitudinal (melodic) and the transversal (harmonic) dimensions. This process is driven by dissonances and consonances among voices that assure both coherence and novelty [1–3].

The polyphonic model is a novel discourse theory in text analysis. Starting from the theories of the Russian philosopher Mikhail Bakhtin [4], this model was created in order to offer a new perspective on understanding how knowledge is built in small groups, to enable the analyze of the interactions among people participating in a conversation and, in general, on how social processes are seen [5].

[1]Eng., Master student, Faculty of Automatic Control and Computers, University "Politehnica" of Bucharest, Bucharest, Romania, (alex.calinescu10@gmail.com).
[2]Corresponding member of AOSR. Prof., PhD, Faculty of Automatic Control and Computers, University "Politehnica" of Bucharest and Senior Researcher, Research Institute for Artificial Intelligence of the Romanian Academy, Romania. stefan.trausan@cs.pub.ro.

Probably the best example of polyphonic music is the fugue, as Johann Sebastian Bach mastered it. In fugues several voices follow diverse counterpoint procedures among one or multiple subjects [6]. Our polyphonic theory of knowledge construction in small groups [1-3, 5] is that successful CSCL conversations follow similar rules to counterpoint in polyphonic music. By sonification we aimed to prove the truth of our theory and the first results, obtained with the MusicXML Creator computer program, orchestrated by Professor Șerban Nichifor from the National University of Music in Bucharest confirmed our assumptions (listen for example to http://www.youtube.com/watch?v=YfuKFdG7ymQ).

The MusicXML Creator computer program was developed for generating a musical composition from a chat conversation. The resulting sonification illustrates how participants interact, how topics of conversation supersede one another, and whether those involved in the discussion contradict or agree on a specific matter. In other words, the sonification emphasizes inter-animation specific to collaborative knowledge construction. In the next section we will present the algorithms for sonification. The MusicXML Creator computer program is presented in the third section of the paper. The fourth section contains.

2. The sonification algorithms

In order to sonify chat conversations, that means, to generate a polyphonic musical piece starting from a chat conversation, several problems should be solved: how to allocate notes to the elements of chats, how voices are allocated to instruments, what is the duration of each note and of rests, and how polyphony is achieved.

For note allocation, we considered two possibilities:

- each participant is a musical note
- selected keywords from the conversation are musical notes.

The musical instruments that will play the generated song are selected at user's preference. The association of voices to instruments is also left at user's choice. Consequently, chat sonification will result in a musical composition with one (on which several voices are played) or with more musical instruments (each instrument being associated to a voice). For each case above mentioned we developed a separate algorithm, the motivation being that for a musical composition with several instruments there must be a different stave filled in simultaneously for each instrument, which makes it difficult to synchronize.

In both cases of note allocation, the duration of a note is determined based on the length of the utterance. In our MusicXML system, the minimum duration of a note is the hundred twenty-eighth note. We chose as minimum duration semiquavers and considered other values as almost imperceptible to the ear.

2.1. The algorithm for the case with one instrument

For computing the duration of a note, we initially considered the interval [minimum length, maximum length] of an utterance, which we divided into 32 equal parts. If the length of the utterance belongs to the first interval, it will be associated with a semiquaver. If it doesn't belong to this range, there will be a new division into 16 equal parts. If it belongs to the first new interval obtained, it will be associated with a quaver. Repeating this step, we reduce the degree of division from 16 to 8, then 4 etc. By reducing the number of divisions, dividing by a factor of 2, the duration of the note is multiplied by the same factor.

We consider this initial form of the algorithm not entirely satisfactory for us because if a person has the habit of talking more, she will be associated with notes with longer duration. For this reason, we have taken the average length of an utterance, and we have taken into account two initial intervals: [minimum length, average length] and [average length, maximum length]. Considering that we wanted the duration of a note to be one of semiquavers, quavers, quarters, minims and semibreves, we chose as average value of them the quarter, which is associated with an interval adjacent to the average length utterance.

We have further changed the way we calculated the length of an utterance after we had noticed the use of emoticons and repeated dots. Therefore, for a more precise calculation of the length of lines (words that are actually used), we considered only the number of alphanumeric characters.

The duration of musical rests is determined by the length of the time elapsed between two utterances. To determine this, we began with a similar "logarithmic" approach to the one used to determine the length of the notes.

There are moments during a conversation in which participants expect a new person to join the chat. These waiting periods strongly influence the values of all the rests that are going to be added to the song. Thus, we changed the initial approach in order to take into account the average response time and standard deviation (σ) of the response time between utterances. However, there were situations where these intervals overlapped due to the high standard deviation. Given that the frequency of notes with long durations was low, we decided to stop using standard deviation and put emphasis on their average.

The next step is to group the sequence of notes and rests into beats. The beat chosen to create the song is 4/4 (equivalent to a semibreve), commonly used in musical compositions.

When the duration of a note is too long compared to the remainder of the beat, it is divided into notes of shorter length, some remaining in the current beat, others being associated with the next beat. In this situation, we don't get the initially

desired effect, to hear a single note of a given duration, but a series of notes identical to the original note having the sum of durations equal to the initial note's duration. When playing music, the notes are sung slightly discontinuous, giving the impression that there were several short utterances instead of a longer one. As a solution, we used a musical tie to continuously sing these notes. An alternative is a "legato", which has the same effect as linking music only used for binding different notes.

2.2. The algorithm for multiple instruments

Determining the duration of a note in the case of several instruments is done exactly as in the algorithm for a single instrument. Figure 1 shows a fragment of the musical composition obtained by applying the initial implementation of the algorithm, without overlapping instruments.

Fig. 1. Initial musical composition fragment.

In the standard chat conversations there are not at least two utterances that overlap in terms of the time they were written at. This type of composition is not polyphonic because it does not contain melodies occurring at the same time, it does not have neither chords. The resulting composition sounds monotone and discontinuous.

To obtain a contrapuntal composition we must synchronize differently overlapping notes belonging to different instruments. When a person makes a reply to an utterance belonging to the same person, the rest between notes is smaller. In this case it is necessary to calculate the following new values : minimum time, maximum time and average time for the response time between utterances belonging to different participants.

Thus, we overlap more notes belonging to different instruments, if the response times between utterances belonging to different participants are less than the average response time between the utterances of different people. If not, musical instruments are synchronized by adding rests to the current maximum total duration of musical elements for an instrument. Whether or not the notes are overlapped, the instruments are then synchronized. Before adding new notes, we need to decide whether to add some rests due to a big response time between the current utterances.

If we want to track in our sonification how certain keywords are used and not to follow participants and if the utterance contains several keywords then associated notes are directly overlapped. Using this method of synchronization between musical instruments, we obtained the fragment shown in Figure 2.

Fig. 2. Musical composition fragment with overlapping instruments.

This type of composition is not polyphonic because it does not contain melodies occurring at the same time. The resulting composition sounds monotone and discontinuous. To obtain a contrapuntal composition we must further synchronize differently overlapping notes belonging to different instruments. When a person makes a reply to an utterance belonging to the same person, the rest between notes is smaller. In this case it is necessary to calculate the following new values: minimum time, maximum time and average time for the response time between utterances belonging to different participants. If the response times between utterances belonging to different participants are less than the average response time between utterances made by different people we overlap more notes belonging to different instruments. If not, musical instruments are synchronized by adding rests to the current maximum total duration of musical elements for an instrument. Whether or not the notes are overlapped, the instruments are then synchronized. Before adding new notes, we need to decide whether to add some rests due to a big response time between the current utterances.

3. The MusicXml Creator System

This section introduces the MusicXml Creator system, its structure, the graphical interface presentation, describing the format of the input files and output mode of association between chat elements and musical elements specific to Music XML format.

The diagram in Figure 3 shows the architecture of the system. It receives an XML file as input (see Figure 4 for an excerpt of such a file). As mentioned in the previous section, the user has the possibility to choose between two combinations: each participant is a musical note or selected keywords are musical note, and musical instruments that will play that song.

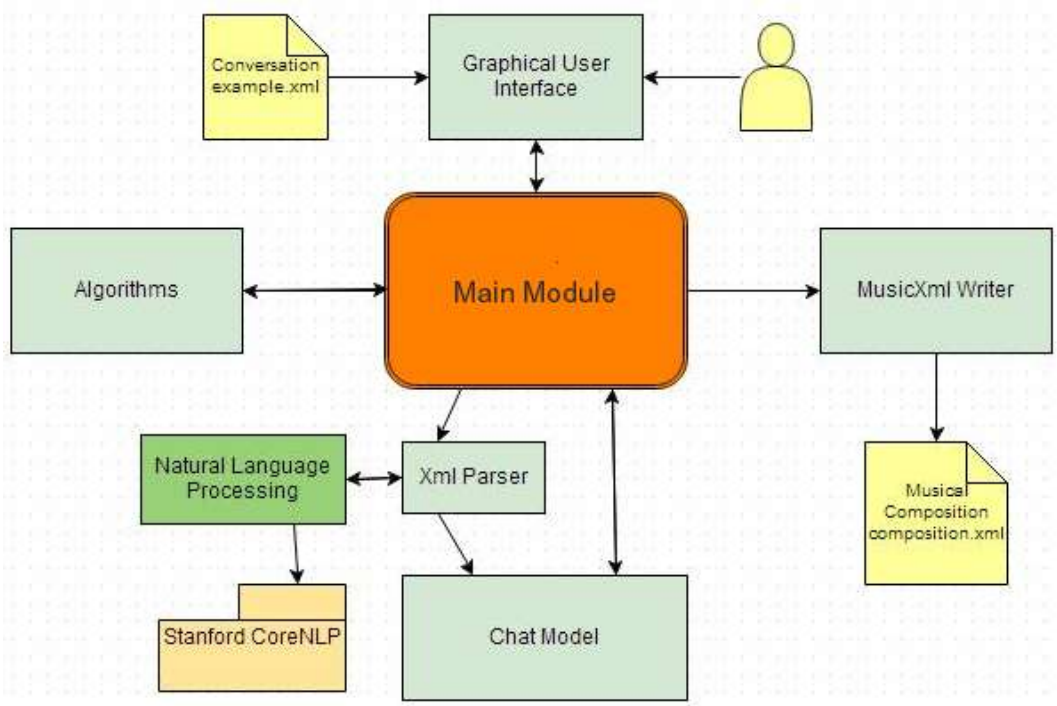

Fig. 3. MusicXml Creator architecture.

The main module takes the input file and parses it using an XML Parser. After this, the natural language text in each utterance of the conversation is processed using a set of modules provided by the Stanford CoreNLP package. Resulting data is stored in the Chat Model (http://www-nlp.stanford.edu/software/).

Data is further taken from the main module and, depending on the selection made in the graphical interface, the appropriate algorithm is called, which sets the duration of the selected notes, their sequence and adds rests where necessary. Finally, notes are grouped into beats and sent back to the main module.

This transmits all data received to the writing module, which generates a Music XML file with the appropriate structure, representing the output file (see Section 3.2). The application input file containing the conversation that is intended to be parsed is an XML file with the structure shown in Figure 4. The participants in the conversation are defined in the beginning of the file. Each individual is characterized by a name (nickname) that is used throughout the conversation.

```xml
<Dialog team="2" file="echipa2.xml">
    <Participants>
<Person nickname="Liviu"/>
<Person nickname="Alex"/>
    </Participants>
    <Topics/>
    <Body>
        <Turn nickname="Liviu">
            <Utterance genid="1" time="03:05:23" ref="0">joins the room</Utterance>
        </Turn>
        <Turn nickname="Alex">
            <Utterance genid="2" time="03:22:56" ref="5">joins the room</Utterance>
        </Turn>
        <Turn nickname="Liviu">
            <Utterance genid="3" time="03:09:05" ref="0">Hey Alex let's make a xml chat example</Utterance>
        </Turn>
        <Turn nickname="Alex">
            <Utterance genid="4" time="03:57:10" ref="3">ok</Utterance>
        </Turn>
        <Turn nickname="Liviu">
            <Utterance genid="5" time="03:57:29" ref="0">Finished</Utterance>
        </Turn>
        <Turn nickname="Alex">
            <Utterance genid="6" time="03:57:54" ref="0">leaves the room</Utterance>
        </Turn>
        <Turn nickname="Liviu">
            <Utterance genid="7" time="03:57:54" ref="0">leaves the room</Utterance>
        </Turn>
    </Body>
</Dialog>
```

Fig. 4. Example of an XML input file.

An utterance is characterized by:

- The participant who emitted it;
- Unique ID (genid);
- The moment when it was emitted (time);
- The utterance’s ID to which reference is made in the text (ref);
- The content of the utterance.

A new person is introduced to the conversation through a line containing the text "joins the room". And when it leaves the conversation, the line will contain the text "leaves the room". If an utterance is not a reply to a previous utterance in the conversation, the ”ref” field will be equal to 0.

3.1. The graphical interface presentation

In the graphical interface, the visual elements are placed in such a way that the user can quickly figure out how to work with it (see Figure 5).

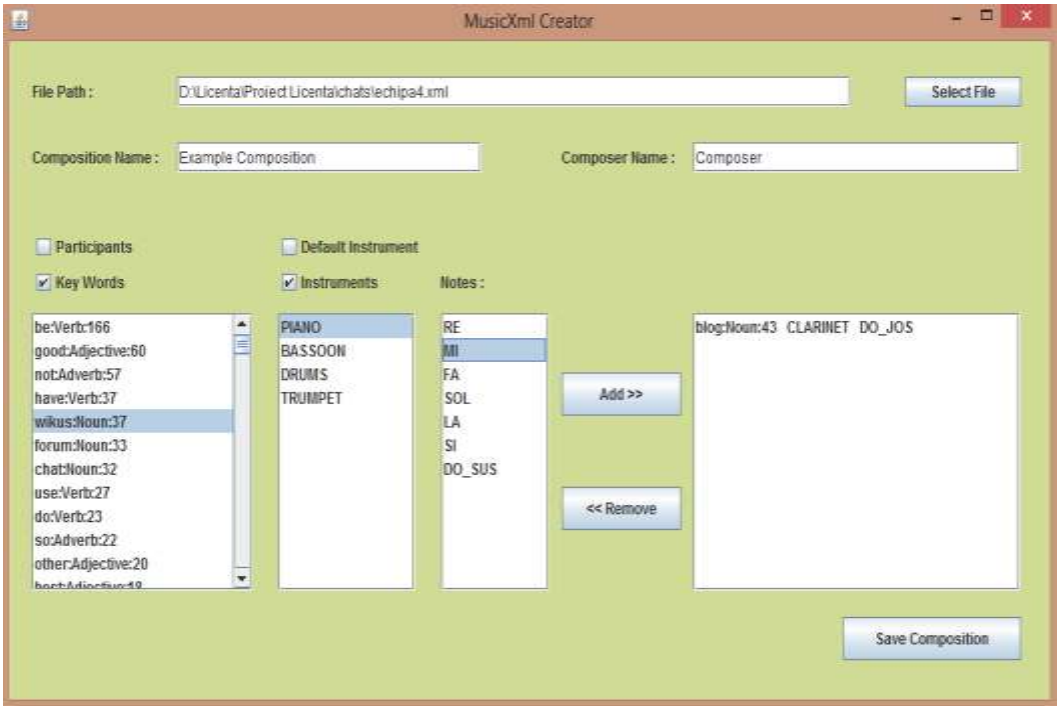

Fig. 5. Graphical User Interface for MusicXml Creator.

Once the input file is selected, the list of keywords or the list of participants to be populated is displayed, depending on the user's choice.

To change the display list corresponding to the first column, uncheck the selected option or select another option (selection of participants or keywords). The second column represents the choice of instruments to be used in playing the resulting song. If "Default Tool" is chosen then all associations will point to piano. The third column lists the musical notes available that can associate a keyword or a participant.

3.2. The MusicXML file obtained

The structure of the MusicXML file created deals with two aspects:

- the visual aspect, which includes the way musical elements, staves, composer's name and the composition's title are arranged (Figure 6);
- the sound aspect, represented by the encoding of musical elements (Figure 7 and Figure 8).

It can be observed that we used the node "identification" to highlight features of the software that helped create the file. The next node is used to set the parameters related to the size of the page, thus facilitating the eventual printing of the musical composition, and, finally, the nodes "credit" used to set a name and author of the composition.

```xml
<identification>
  <encoding>
    <software>MusicXml Creator</software>
    <supports attribute="new-system" element="print" type="yes" value="yes"/>
    <supports attribute="new-page" element="print" type="yes" value="yes"/>
  </encoding>
</identification>
<defaults>
  <scaling>
    <millimeters>7.2319</millimeters>
    <tenths>40</tenths>
  </scaling>
  <page-layout>
    <page-height>1545</page-height>
    <page-width>1194</page-width>
  </page-layout>
</defaults>
<credit page="1">
  <credit-words default-x="600" default-y="1490" font-size="24" justify="center" valign="top">Multi-instrument Composition</credit-words>
</credit>
<credit page="1">
  <credit-words default-x="1125" default-y="1410" font-size="12" justify="right" valign="top">Calinescu Alexandru</credit-words>
</credit>
```

Fig. 6. Fragment of created MusicXML file – visual aspect.

Music XML file structure related to the sound is based on two main nodes:

1) “Part-list”, which includes a listing of all the instruments and their association with corresponding parts. An instrument is represented by a node "score-part" having the structure shown in Figure 7.

```xml
<score-part id="P1">
  <part-name print-object="yes">Piano</part-name>
  <score-instrument id="P1-I1">
    <instrument-name>None</instrument-name>
  </score-instrument>
  <midi-instrument id="P1-I1">
    <midi-channel>1</midi-channel>
    <midi-program>1</midi-program>
    <volume>80</volume>
    <pan>0</pan>
  </midi-instrument>
</score-part>
```

Fig. 7. Fragment of created MusicXML file – defining musical instruments.

It can be seen that each instrument is characterized by a part id (id = "P1"), a midi instrument id seen as an independent musical device for the MIDI protocol used for song playback, an unique channel playback (midi-channel), an inner coding id instrument and traits related to the sound produced by the instrument. 2) “Part”, which contains the score associated with an instrument. This includes a list of nodes "measure" as a beat which in turn, contains a list of nodes "notes" that represent a note or a musical rest. The structure of a "note" node is shown in Fig. 3.8.

```
<note>
  <rest measure="yes"/>
  <duration>64</duration>
  <voice>1</voice>
</note>
<note>
  <pitch>
    <step>C</step>
    <octave>5</octave>
  </pitch>
  <duration>8</duration>
  <voice>1</voice>
  <type>16th</type>
  <stem>down</stem>
</note>
```

Fig. 8. Fragment of created MusicXML file – defining musical elements.

The first "note" node represents a musical rest with the second one representing a musical note. A common feature of the two notes is their duration (the notes differ because of the concept of octave). Other features are: the position of the element in the octave, duration in units, type and how the way to draw the note. A unit is associated with a hundred twenty-eighth notes.

4. Testing and evaluation

In order to test the application, we used music arrangements that include notes of counterpoint compositions. These are characterized by original notes and cadences given in Table 4.1.

Mode	Initial Mode	Frequent Cadence
Dorian	Re, La	Re, La, Fa
Phrygian	Mi, La, Si	Mi, La, Sol
Lydian	Fa, Do	Fa, Do
Mixolydian	Sol, Re	Sol, Re, Do
Aeolian	La, Mi	La, Re, Do
Ionian	Do, Sol	Do, Sol, La

Table 1. Musical Modes.

The choice of the music arrangements has been made in order for the resulting composition to sound as harmonic as possible. To illustrate rhythmic response of two participants in a conversation we used notes from the Lydian mode (Figure 9).

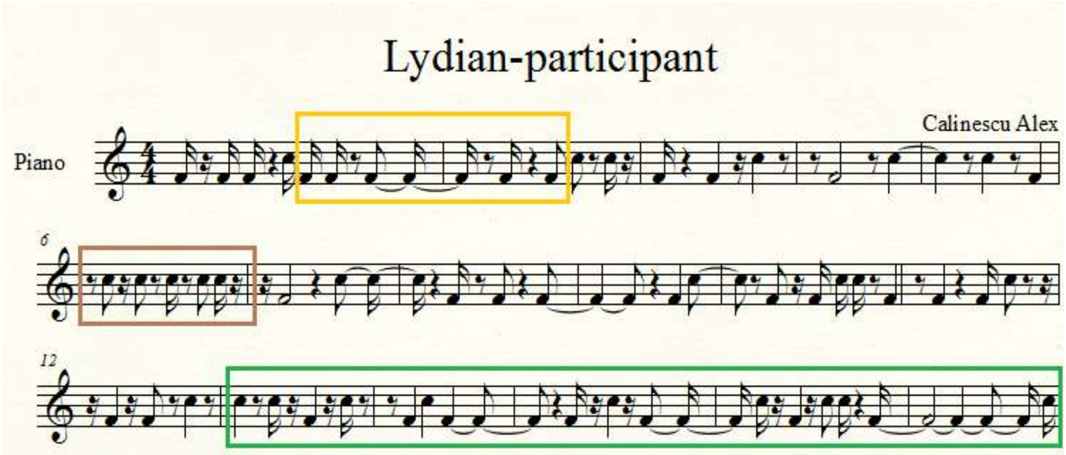

Fig. 9. Musical composition fragment – selected participant, Lydian mode.

The yellow highlighting indicates a sequence of utterances belonging to the participant who has been associated with the note “Fa”.The brown highlighting contains a series of utterances belonging to the participant who has been associated with the note “Do”. The green highlighting indicates the alternation of utterances of participants, suggesting a communication of "request-response".

To generate a sonification that allows analyzing the interactions of keywords in the conversation, in the example shown in Figure 10, we chose a musical arrangement of notes used in the Phrygian mode. We can see a frequent usage of keywords in several fragments, which usually imply that the topic of discussion is prompted by the word. We associated the note “Mi” with the keyword "chat", topic highlighted by the green box and the note “La” with the keyword "forum".

Fig. 10. Musical composition fragment – keywords selection, Phrygian mode.

In order to observe the usage of certain words in a conversation (in a negative or positive context) , in the example in Figure 11, we associate the adverb "not" with the note “Do”, the verb "agree" with the note “Sol” and the adjective "good" with the note “La”. The green box highlights when there is a dispute between participants.

Fig.11. Musical composition fragment – highlighting positive and negative context.

The above examples were made choosing "Default Instrument", seeking harmony of sounds made by the chosen music arrangements and the repetitive fragments in order to determine the existence of patterns in the way participants interact or in the way they alternate topics.

With the musical composition played by many instruments we want to analyze how they overlap in order to understand which topics are discussed at a certain point in time or how the participants are involved. The overlap of instruments is represented by a red line in the example shown in Figure 12.

Fig. 12. Musical composition fragment – overlapping instruments.

In the following example we have highlighted a fragment where a participant is not sufficiently engaged in the conversation, preferring to follow what others discuss. This participant is associated with the trumpet instrument, and his period of inactivity is evidenced by the series of rests in the green box (Figure 13).

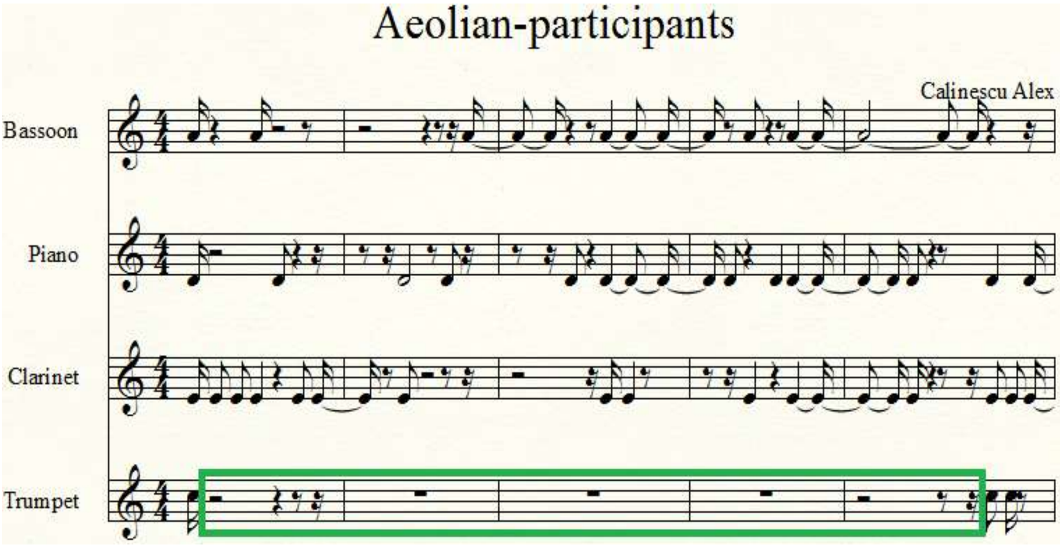

Fig.13. Musical composition fragment – insufficient participant engagement.

The fragment in Figure 14 shows the use of the words "good" and "yes" to which we associate the corresponding high-pitch notes, respectively top “Do” and “Si” and the use of the words "not" and "problem" to which we associate the bass notes, respectively bottom “Do” and “Re”.

Fig. 14. Musical composition fragment – highlighting positive context.

We can observe there is a stave area where the frequency of the word "good" is high. This fragment is played by an instrument which emphasizes the rhythm of participants when it comes to agreeing with the words of another colleague. In this case, the instrument that plays the note “Si” also has a low accompaniment from instruments assigned to stave one, three and four.

In terms of sound, in these situations it is advisable to associate positive words with instruments such as the clarinet and bassoon (which have a higher playback frequency range) and for those with negative aspect instruments such as drum or trumpet.

In all tests performed, we chose words that have a high frequency of usage. If we had used words with a low frequency of occurrence, we would not have obtained a musical composition representative for our study.

For a correct understanding of the rhythm of the conversation, it is advised to select all the participants.

Musical compositions with several instruments have the advantage of the possibility of eliminating a participant or a word, by disabling an instrument.

Starting from the results generated by the system, an orchestration was performed by Professor Șerban Nichifor from the National University of Music in Bucharest and the resulted musical pieces were beyond our expectations, for example, the 3 Dances musical piece, which integrated three chat sonifications and can be listened at http://www.youtube.com/watch?v=YfuKFdG7ymQ.

5. Conclusions and future developments

The association between an utterance of a conversation and a musical note is difficult to implement; choosing the note depends on the message that is sent and the tone used, aspects that are difficult to extract from a chat conversation.

This application achieves its purpose based on the results obtained after a series of tests on chat conversations.

Although the songs created are not masterpieces of art developed by composers and musicians, we believe that, for a person who does not have advanced musical knowledge, they actually seem to reflect the messages that are intended to be transmitted by those involved in the conversation.

In conclusion, an audio representation of a chat conversation is a complex process influenced by many factors that must be taken into account in order to get the most accurate rendering of ideas and moods of the participants. However, this does not prevent us to believe in a future "artistic maturity" of the computer, which will transform ordinary chat conversations into veritable symphonies of information.

Acknowledgements

We would like to kindly thank to Professor Șerban Nichifor from the National University of Music in Bucharest for his enthusiasm, encouragements, comments and corrections on previous versions of the paper, and for the extremely precious work on orchestrating and disseminating our sonifications. Special thanks should be directed to Dr. Hardy F. Schloer for his encouragements in approaching sonfication, which were determinant for the present work.

R E F E R E N C E S

[1] Stefan Trausan-Matu, *The Polyphonic Model of Hybrid and Collaborative Learning*, In F. Wang, L., J. Fong. & R. C. Kwan (Eds.), Handbook of Research on Hybrid Learning Models: Advanced Tools, **2010.**

[2] Stefan Trausan-Matu, Stahl, G., & Sarmiento, J. *Supporting polyphonic collaborative learning*, Indiana University Press, E-service Journal, 6(1), 58–74, **2007.**

[3] Ștefan Trăușan-Matu, *Studiu asupra perspectivei polifonice în analiza discursului în texte, în conversații și în muzică*, Raport de cercetare ICIA, iunie **2013.**

[4] Mikhail Bakhtin, *The dialogic imagination: Four essays* (C. Emerson & M. Holquist, Trans.), Austin and London: The University of Texas Press, **1981.**

[5] Stefan Trausan-Matu, *A Polyphonic Model, Analysis Method and Computer Support Tools for the Analysis of Socially-Built Discourse,* Romanian Journal of Information Science and Technology 2-3 (16), pp. 144-154, **2013.**

[6] Liana Alexandra Moraru, *Analize Polifone Tonale*, Editura Universității Naționale de Muzică, București, **2006**.